

SYSTEM AND METHOD FOR BLIND MEDIA SUPPORT

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosures, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

RELATED APPLICATIONS

This application claims priority to U.S. provisional application number 60/411,267 filed on September 16, 2002, the entirety of which is hereby incorporated by reference.

This application is related to the following pending applications, each of which is hereby incorporated herein by reference in its entirety:

- Application Serial No. 09/495,751, titled HIGH SPEED DATA TRANSFER MECHANISM, filed January 30, 2000, attorney docket number 4982/7;
- Application Serial No. 09/354,063, titled MODULAR BACKUP AND RETRIEVAL SYSTEM, filed July 15, 1999, attorney docket number 4982/4;
- Application Serial No. 09/354,058, titled HIERARCHICAL BACKUP AND RETRIEVAL SYSTEM, filed July 15, 1999, attorney docket number 4982/5;
- Application Serial No. 09/610,738, titled MODULAR BACKUP AND RETRIEVAL SYSTEM USED IN CONJUNCTION WITH A STORAGE AREA NETWORK, filed July 6, 2000, attorney docket number 044463-002;

Application Serial No. 09/609,977, titled MODULAR BACKUP AND RETRIEVAL SYSTEM WITH AN INTEGRATED STORAGE AREA FILING SYSTEM, filed July 5, 2000, attorney docket number 044463-0023;

Application Serial No. 09/774,268 titled LOGICAL VIEW AND  
5 ACCESS TO PHYSICAL STORAGE IN MODULAR DATA AND STORAGE MANAGEMENT SYSTEM, filed, January 30, 2001, attorney docket number 044463-0035;

Application Serial No. 09/038,440, titled HIGH-SPEED DATA TRANSFER MECHANISM, filed March 11, 1998, attorney docket number 044463-  
10 0002; and

Provisional Application Serial No. 10/303,640 titled SELECTIVE DATA REPLICATION SYSTEM AND METHOD, filed November 23, 2001, attorney docket number 4982-21PROV.

## BACKGROUND OF THE INVENTION

### 15 Field of the Invention

The invention relates to data storage in a computer network, and more particularly, to data storage supporting a blind library.

### 20 Description of Related Art

Removable media storage devices such as tape libraries, optical libraries, and CD-ROM jukeboxes are frequently used to backup data in computer networks. A removable media storage device generally includes removable media such as tape cartridges, optical

disks, or CD-ROMS and a mechanism, such as a robotic arm, that moves the removable media into and out of a drive where content of the media is read or updated.

The removable media are identified in a variety of ways ranging from hand printed labels to machine readable bar codes – all disposed tangibly on an external portion of the media. The media identification can indicate which media is used to store particular data. For example, data from a particular network source, such as a computer or a database, may be designated for storage on a specific tape cartridge. The identification on the tape permits selection of the tape for storage and retrieval of the data stored thereon.

The use of bar codes in removable media identification allows removable media storage libraries to operate autonomously and with greater efficiency. The GALAXY data storage management system software manufactured by COMMVault Systems, INC. of Oceanport, New Jersey is an example of a storage system that uses bar coded removable media to perform autonomous storage operations. GALAXY uses removable media storage devices equipped with bar code readers to select removable media for storage operations based on the respective bar codes. A library with a bar code reader automatically identifies a specific removable media according to the bar code thereon and inserts the removable media in a corresponding drive to read or update content. User input is not necessary at each instance and storage operations can proceed according to predefined schedules and criteria resulting in greater efficiency and ease of use than if user input were needed.

Removable media without bar codes and/or storage libraries without bar code readers, however, do not permit such autonomous storage operations. Such libraries are

known as blind libraries since conventional storage management software is unable to identify such removable media and user interaction is needed. This inconveniences users and results in decreased efficiency of network storage operations. Yet, many removable media libraries are not equipped with bar code readers and provide no alternative to such manual user input without replacing the entire library.

Thus, there is a need in the art for a system and a method that supports blind libraries in network storage operations.

#### SUMMARY OF THE INVENTION

The invention generally provides methods and systems for detecting sighted and blind network storage libraries, identifying media contained in the network storage libraries, updating a storage management database with information regarding the network libraries and media, monitoring network libraries for state changes that could indicate the addition or removal of storage media, and updating the storage management database with information regarding the state changes of the storage media.

In the case of sighted libraries, the invention scans bar codes or other machine-readable information to identify each piece of media. In the case of blind libraries, the invention loads each piece of media into a library drive and reads an on-media label (“OML”) directly from the media itself. As further described herein, the OML is a unique identifier written as data on the removable storage media.

The invention includes a system and method for operating a storage device. The system comprises a management server, a media agent connected to the management server, a storage device connected to the media agent and a database connected to the

management server. The management server controls the media agent to monitor for the addition or removal of a piece of media in the storage device. When the media agent determines that the piece of media has been added to the storage device or removed from the storage device, the media agent causes the storage device to read a media label stored  
5 as data on the piece of media, the media label including an identifier identifying the piece of media. When a backup or restore is performed, the system checks the media label to ensure that the correct media is being used. Inventory procedures are disclosed. Media labels are stored in a database along with an indication of the corresponding slot where the media is stored, a time when the label was put on the medium, and a confidence  
10 parameter indicating how confident the system is that the particular medium is stored in the respective slot.

One aspect of the invention is a system for operating a storage device, the system comprises a management server, a media agent connected to the management server, a storage device connected to the media agent, and a database connected to the  
15 management server. The management server controls the media agent to monitor for the addition or removal of a piece of media in the storage device. When the media agent determines that the piece of media has been added to the storage device or removed from the storage device, the media agent causes the storage device to read a media label stored as data on the piece of media, the media label including an identifier identifying the piece  
20 of media.

Another aspect of the invention is a system for operating a storage device, the system comprising a management server, a media agent connected to the management server, a storage device connected to the media agent, and a database connected to the

management server. The management server controls the media agent to monitor for the addition of a piece of media in the storage device. When the media agent determines that the piece of media has been added to the storage device, the media agent causes the storage device to write a media label stored as data on the piece of media, the media label  
5 including an identifier identifying the piece of media.

Yet another aspect of the invention is a system for backing up data in a storage device, the system comprising a management server, a media agent connected to the management server, a storage device connected to the media agent, and a database connected to the management server. The storage device loads a piece of media. The  
10 storage device stores a media label as data on the piece of media, the media label including an identifier identifying the piece of media. The media agent transfers backup data to the piece of media.

Still yet another aspect of the invention is a system for transferring data between a data source and a desired piece of media, the system comprising a management server, a  
15 data source connected to the management server, a media agent connected to the management server, a storage device connected to the media agent, and a database connected to the management server. The storage device loads a test piece of media. The storage device reads a media label stored as data on the test piece of media, the media label including an identifier identifying the test piece of media. The media agent  
20 transfers data between the data source and the test piece of media when the media label corresponds to the desired piece of media.

Another aspect of the invention is a storage device system comprising a management server, a media agent connected to the management server, a storage device

connected to the media agent, and a database connected to the management server. The storage device includes a plurality of pieces of media, each piece of media including a respective media label stored as data on the respective piece of media, each media label including a respective identifier identifying the respective piece of media.

5           Yet another aspect of the invention is a method of performing an inventory of media stored in a storage device system, the storage device system including a management server, a media agent connected to the management server, a storage device connected to the media agent, and a database connected to the management server. The method comprises loading a piece of the media stored in a slot of the storage device into  
10 a drive. The method further comprises reading a media label stored as data on the piece of media, the media label including an identifier identifying the piece of media. The method further comprises storing the identifier in the database in association with an indication of the slot.

          Still yet another aspect of the invention is a computer readable storage medium  
15 including computer executable code for enabling a system to transfer data between a data source and a desired piece of media, the system comprising a management server, a data source connected to the management server, a media agent connected to the management server, a storage device connected to the media agent, and a database connected to the management server. The computer readable storage medium including data for  
20 performing the steps of: controlling the storage device to load a test piece of media. The computer readable storage medium further including data for controlling the storage device to read a media label stored as data on the test piece of media, the media label including an identifier identifying the test piece of media. The computer readable storage

medium further including data for controlling the media agent to transfer data between the data source and the test piece of media when the media label corresponds to the desired piece of media.

5

## BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 is a block diagram of a network architecture in accordance with one embodiment of the invention; and

Fig. 2 is a flow chart showing a method in accordance with one embodiment of the invention.

10

Fig. 3 is a flow chart showing a method in accordance with one embodiment of the invention.

Fig. 4 is a flow chart showing a method in accordance with one embodiment of the invention.

15

## DETAILED DESCRIPTION OF THE INVENTION

Referring to Fig. 1, a storage library system 200 in accordance with the invention comprises a storage management server 100 including a media library manager module ("MLM") 105. A storage management database 110 is connected to MLM 105. A media agent 115 provides communication between server 100 and a removable media storage device library 120. Removable media 125 are used for storage and retrieval of network data stored thereon. Removable media 125 includes, for example, a plurality of tapes stored in slots 126. Although only one media agent 115 and library 120 are shown, clearly a plurality of media agents and libraries may be connected to a single

20



management server. Software for running some operations of server 100 may be stored on a medium 102.

As described in Application Serial No. 09/354,058, titled HIERARCHICAL BACKUP AND RETRIEVAL SYSTEM, filed July 15, 1999, attorney docket number 4982/5, and other applications incorporated by reference herein, modules of the storage management server 100 coordinate and direct storage operations for system 200. For example, media library manager module 105 on storage management server 100 is responsible for communicating with media agents 115 to perform all media related operations. Media library manager module 105 operates according to preferences and data stored in storage management database 110. For example, media library manager module 105 might instruct media agent 115 to periodically backup network data from a particular source 101 to a specific one of removable media 125 in library 120. Storage management database 110 tracks, among other things, network data sources, available removable storage media, such as an individual tape 125a, and related associations among these items.

Library 120 may include blind and/or non-blind or “sighted” media 125. System 200 tracks all known media available for network storage operations including when medium 125a is added or removed from storage library 120. In the case of “sighted” media and libraries, system 200 automatically identifies when new medium 125a with a bar code or other machine-readable identification is added to library 120 by scanning the bar code and updating storage management database 110 accordingly. A bar code, for example, might comprise a combination of letters and numbers or a group of lines symbolizing such an alpha-numeric combination. This alpha-numeric combination on

medium 125a corresponds to an entry 112 in a library slot table 116 of storage management database 110. In library slot table 116, entry 112 is associated with other media-related information 114 indicating, for example, the storage capacity of library 120, an index of the data stored on individual medium 125a, and other useful information.

System 200 automatically identifies when a new piece of media 125 is added or removed from a blind library using additional techniques as further described herein.

Referring also to Fig. 2, when system 200 is first installed or at other times as instructed by a user or otherwise scheduled, network media agents 115 determine which removable storage media device libraries 120 are blind libraries and which libraries 120 are sighted libraries, at step S130 etc. A dedicated library and drive configuration software tool may be used to detect blind and sighted libraries instead of, or in conjunction with, network media agents 115. The library type information is stored in storage management database 110, at step S135 etc. Media agents 115 monitor network libraries 120 for state changes indicating the addition or removal of a piece of media 125, at step S140 etc. For example, when a library door is opened or a library mail slot is accessed, this may indicate that medium 125a has been added to or removed from library 120.

When the media agent 115 detects the addition or removal of a new piece of media 125 from the removable storage media device library 120, the media library manager 105 is informed and consults the storage management database 110 to determine if the library 120 is a blind or a sighted library, at step S145 etc. If the library 120 is a

sighted library, the library 120 scans the bar code identifier on the appropriate medium 125a, at step S150 etc., and passes this information to the media agent 115.

If library 120 is a blind library, library 120 physically loads the appropriate piece of media 125 into a drive and reads an on-media label (“OML”) stored as data in piece of media 125, at step S155 etc., and passes this information to media agent 115. The OML is a unique identifier written as data on the actual medium 125a. An OML might comprise an alpha-numeric combination. This alpha-numeric combination on medium 125a corresponds to entry 112 in table 116 of storage management database 110. Entry 112 is associated with other media-related information indicating, for example, the storage capacity, slot location in library 120, an index of the data stored on medium 125a, and other useful information.

The media agent 115 passes the information identifying medium 125a to media library manager module 105, and storage management database 110 is updated, at step S160 etc., to indicate the medium 125a present in the library 120. Control returns to step 140 and the media agent 115 continues monitoring state changes of the library 120.

A library management service (“LMS”) running on the media agent 115 may be used to support blind libraries in the network. The LMS is a process that monitors the library 120 for state changes and signals MLM 105 when the LMS detects the door of library 120 opening or closing, or when the mail slot is accessed. The LMS also issues mount and un-mount commands to library 120 to facilitate storage of data from a network source to a particular media or destination. These associations may be specified by a storage policy, by a user, or by other means, and is stored in the storage management database 110. The LMS also produces a slot configuration for all slots that are occupied.

System 200 supports different forms of media identification for blind libraries. Library slot table 116 in storage management database 110 tracks specific information about medium 125a stored in network libraries 120. A confidence parameter 118 is stored in slot table 116 indicating how confident MLM 105 is about the identity of a

5 medium 125a in a particular library slot 126. Slot table 116 may, for example, identify the location of a particular medium 125a in a particular slot with the values confident, not-confident, not-known, and inventory in progress. If there is an entry in table 116 which indicates that medium 125a is in slot 126a and the confidence parameter is set to confident, then MLM 105 is sure that slot 126a has medium 125a. Not-confident slots

10 are those slots that were previously marked confident, but now have become not-confident because a state change was detected - such as the library door opening or the mail slot being accessed. Unknown slots are those slots that have recently been occupied by media but MLM 105 has not read the OML on the media 125 in these slots. Slot table 116 also stores timestamps 122 indicating the last inventory update time for a library slot

15 and a time stamp when the last inventory was done.

System 200 detects and identifies medium 125a stored in library 120 using a number of different operations including a forced-full inventory operation, an update inventory operation, a search inventory operation. Inventories generally have the lowest priority.

20 A forced-full inventory operation involves reading the OML of all removable storage media 125 in library 120. The forced-full inventory operation is generally initiated by a user, but can also be automatically initiated by MLM 105. This operation is often very lengthy and an appropriate warning may be posted to the user with alerts and

events. To initiate a forced-full inventory, the MLM 105 populates an administrative task schedule table with the appropriate information or otherwise commences the forced-full inventory operation. A dedicated Application Interface ("API") then starts the inventory process that performs the forced-full inventory. At the end of a forced-full inventory, the inventory process calls an MLM 105 API to indicate the end of the inventory process.

After the forced-full inventory operation is complete, all the slots will have the confidence parameter set to confident and the library slot table 116 in the storage management database 110 is updated accordingly. If the library door is opened or the mail slot accessed during the operation, the entire operation is cancelled. Alternatively, library 120 may be marked offline while a forced-full inventory operation is in progress. The forced-full inventory operation should not be performed when backups/restores are currently running on a library 120. Once the forced-full inventory is complete, all known media not found in the library are marked as exported. The forced-full inventory will automatically discover new media because all media in slots are inventoried.

The update inventory operation is a user triggered or MLM 105 triggered incremental inventory. This operation is started so that slots that have not been inventoried or slots that have confidence level set to not-known are inventoried to update the confidence level. Slots that are marked non-confident are generally left alone and non-confident slots are marked as confident when the medium 125a from that slot is mounted for backups or restores as further described herein. Library 120 is generally not marked offline during update inventory process. The update inventory operation may become a forced-full inventory operation if all the library slots are marked non-known. The MLM 105 may start an update inventory operation the first time library 120 is

configured, migrated from another storage management system, or whenever media agent  
115 services are restarted.

Once the user introduces a new or used medium into a library, the system starts an  
update inventory operation which will inventory slots that are unknown as described  
5 herein.

A search inventory operation searches for a particular removable storage medium  
125a or set of media 125 in library 120. The search inventory operation is often started  
when a mount fails and the MLM 105 service needs to locate particular medium 125a.  
The MLM 105 populates an administrative task table with a list of the media 125 that  
10 needs to be located. Multiple drives may be used in parallel during a search operation. If  
the inventory process dies in the middle of a search, the job requiring the search is  
terminated.

During inventory operations, the inventory process calls a media manager API to  
read each OML. This API issues a system wide blocking call that will reserve a drive and  
15 sets the slot confidence level to identification in progress. The media manager API  
mounts the medium 125a in the library 120 slot, reads the OML, updates the storage  
management database 110, and returns to the inventory process. The inventory process  
may be multi-threaded and utilize all the drives that are free for the inventory process. A  
drive that was reserved by an application but failed to mount is flagged as usable.

20 As discussed, the OML written on the media itself is used to identify removable  
storage medium 125a. MLM 105 uses the storage management database 110 to keep  
track of which slot 126 includes discovered media 125 so that MLM 105 can perform  
future mounts of requested medium 125a without searching for them. If a required

medium 125a, however, is not present in the slot indicated by the storage management database 110, then the required medium 125a should be searched for in the library 120. The system uses a number of different search algorithms to locate media.

5       A pigeon hole search scheme is used when a medium X is required for a job. For example, assume that table 116 indicates that medium X is in a particular library 120 in slot 126a. When mounted, the MLM 105 discovers that the mounted medium in slot 126a is not X but Y, and Y was listed as being in slot 126b. MLM 105 updates the library slot table 116 for medium Y and indicates that the medium in slot 126a is Y and not X. Then MLM 105 mounts the medium in slot 126b to test if it is X. If the medium mounted from  
10   slot 126b is indeed X then the storage management database 110 is updated and the job continues. If not, the MLM 105 identifies the medium that was mounted from slot 126b, updates the storage management database 110 entries for that medium, determines the original location where this medium was stored and mounts the medium from that slot as discussed above. This process continues until medium X is found.

15       Another searching algorithm is called the oldest mounted media search algorithm, and operates as follows. When a slot is marked as confident, the time is updated in field 122 of table 116 indicating when the slot was marked confident. This time field is used to search for a desired medium 125a in the slots 126 that are marked confident if a desired medium 125a was not found in any other slots. The slot with the oldest update  
20   time will be searched first. Unknown slots may have a timestamp of 0.

A library may be kept online when an inventory is in progress except when during a forced-full inventory. Alternatively, at any given time, only one inventory process can

be running for a library and the information of a running inventory process and the type of inventory is stored in table 116 and displayed on a GUI.

A summary of copy and restore operations of the invention will now be explained.

Referring to Fig. 3, if a copy or storage operation job requests MLM 105 to mount a  
5 piece of media 125a, step S202 the system determines whether the medium 125a is an  
unknown medium at step S204. If the medium is unknown, system 200 writes an OML  
on medium 125a, step S206. This OML identifies this medium 125a in future operations  
within system 200. Once the OML is written on medium 125a, system 200 also updates  
library slot table 116 in the storage management database 110, identifying medium 125a,  
10 setting medium 125a as discovered, and indicating the time the OML was written on to  
the medium 125a, step S208. The corresponding slot is also listed in slot table 116 and  
indicated as being confident. Thus, new undiscovered medium 125a in the library 120 is  
discovered and the necessary information updated in the storage management database  
110. If the medium 125a mounted is not a spare or unused media, then MLM 105 will  
15 look for the next spare media for the backup, step S210.

If the medium 125a to be mounted has already been discovered or is known at  
step S204, then MLM 105 reads the OML on the medium 125a to validate and ensure  
that the correct media has been mounted, step S212. If the medium mounted in the drive  
is correct then the copy continues. If the mounted medium is not the correct one, then the  
20 correct medium should be searched for and mounted, step S214. The medium is then  
searched in the library 120 using one of the search schemes described above. A mount  
API indicates success or failure of the search operation. The system handles errors and  
retries until successful, until a different error code is returned, or the system passes an



error flag to an API to indicate that it should wait and then retry in case the mount failures are due to an incorrect medium in a slot.

If the mounted medium is a new spare medium, then system 200 erases any existing OML, writes a new OML, and updates the storage management database 110. If the medium 125a belongs to a spare group identified in the storage management database 110, that is different from a desired spare group, then the medium 125a is moved into the correct spare group corresponding to the media group. This reduces mount time. If the medium 125a is not found in the library 120 then the medium 125a is marked full and the backup continues on the next confident spare media. If there are no confident spare media 125, then the next not-confident spare media is selected as the active medium.

An incremental inventory may be started to search for a spare media 125 that was not found. Alternatively, the list of media 125 that needs to be searched for can be updated in an administrative task table even when an inventory process is already searching for a given medium 125a. When allocating media to a media group, confident media may be considered first.

Referring to Fig. 4, restore jobs retrieve specific data stored on a specific medium 125a. System 200 mounts the required medium 125a from the slot 126 where the storage management database 110 indicates the medium 125a is residing, step S222. The OML of the medium 125a is read to verify that the correct medium 125a has been mounted, step S224. If the OML is correct, the restore operation continues and data is retrieved from the medium 125a, step S226. If the OML is incorrect indicating that the wrong medium 125a is in the drive, the correct medium should be searched for in the library 120 before the job can continue, step S228. System 200 may inform the user that the

requested medium 125a is not in the library 120. Alternatively, MLM 105 or storage management database 110 may indicate that the medium 125a was exported, and the system 200 displays a message to the user indicating that medium 125a is outside the library 120. Another option is for the user to manually force a search inventory operation  
5 to try and locate the missing medium.

Confident media can be exported without verification. Media in slots 126 that are not confident should be verified before export. Exports are scheduled or initiated manually by a user. A message prompting the user to label the media may be shown on a GUI from the invoked export. The export process calls a media manager API to verify  
10 the OML and then export the media. Export media verification is performed sequentially. Once all the media have been verified, then all the media is exported. Users can also select a specific media to export by slot without verification. For example, a user may want to remove a medium and does not want it to be verified. Appropriate warnings may be issued when users attempt to export media without verification.

15 If the connection between storage management server 100 and media agent 115 is lost, storage management server 100 attempts to reconnect to media agent 115. If the connection cannot be reestablished, then media agent 115 is presumed dead and all of the slots for all libraries 120 connected to and controlled by this media agent 115 are marked as not-confident.

20 A user may be provided with an easy upgrade path to migrate an RSM or other storage management system controlled library to the blind library control system of the present invention. This is supported regardless of whether library 120 is blind or sighted. The process involves resetting a number of fields stored in the storage management

database 110 including the GUID field on all the Media, MediaGroup, SpareGroup, Library, and Drive tables and updating the drive controller and library controller types to DMS/LMS controlled. The slot table entries for the library are deleted. The LMS recreates the slot table entries and all the slots are marked as unknown and an update  
5 inventory is started for a blind library.

A blind library can be converted to a sighted library by labeling all the media 125 with a unique barcode and fitting a barcode reader in the library. A library and drive configuration tool allows the user to change the property of a library from blind to sighted in the storage management database 110. A user might, for example, use a GUI  
10 or other input method to input changes to the storage management database 110 and update fields to change the property of a library 120. Once a barcode reader has been fitted, the user re-configures the library as a sighted library. A flag is set that indicates that MLM 105 has to validate the barcode reader and co-relate the barcode pushed up by the LMS to all the media 125 that are currently in the slot table in the storage  
15 management database 110.

If a library inventory is started and all the slots for a library return a media status set to invalid barcode, then the presence of barcode reader has not been verified. Common causes are that the user made a mistake about the library being sighted or the barcode reader is not functioning correctly. In either case an event will be logged in the  
20 storage management database indicating the problem and the library will be marked offline. The user will then have a choice to mark the library back to blind.

Once the barcode reader has been verified, MLM 105 will start a full-forced inventory process on the library whereby it will mount all the media in the library, read

the OML for the media 125, and update the slot table 116 with the correct media identifier. When all the slots have been verified, then the library 120 is ready to use. Once this process completes, a barcode matching flag is reset and the library 120 becomes permanently sighted.

5 Media 125 that has been exported from library 120 should be updated with the barcode. A user may employ the unique Ids in the storage management database 110 to correlate the media 125. The barcode fitted on each of these media 125 is then stored in storage management database 110. Users may enter barcode information via a GUI.

If the users skip the GUI step, then the media 125 (now fitted with barcodes) are  
10 imported into the library 120 and are treated as spares. When a spare is allocated and mounted, if it is found that it is a medium 125a with valid data, then the storage management database 110 is updated accordingly. A fresh spare is then selected and the operation continues.

If the user has made a mistake in associating barcodes, (or barcodes were  
15 swapped on an already sighted library 120) then during a mount, MLM 105 will detect a medium 125a whose OML doesn't match with the barcode. An appropriate event is generated citing the cause of failure and listing both the barcodes. A user intervention is then required to correct the situation by updating the barcodes from the GUI.

Thus, by providing an on media label stored as data on the piece of media itself, a  
20 system is created where blind media library may operate autonomously.

Systems and modules described herein may comprise software, firmware, hardware, or any combination(s) of software, firmware, and/or hardware suitable for the purposes described herein. Software and other modules may reside on servers,

workstations, personal computers, computerized tablets, PDAs, and other devices suitable for the purposes described herein. Software and other modules may be accessible via local memory, via a network, via a browser or other application in an ASP context, or via other means suitable for the purposes described herein. Data structures described herein

5 may comprise computer files, variables, programming arrays, programming structures, and/or any electronic information storage schemes or methods, or any combinations thereof, suitable for the purposes described herein. User interface elements described herein may comprise elements from graphical user interfaces, command line interfaces, and other interfaces suitable for the purposes described herein.

10 While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications as will be evident to those skilled in this art may be made without departing from the spirit and scope of the invention, and the invention is thus not to be limited to the precise details of methodology or construction set forth above as such variations and modification are intended to be

15 included within the scope of the invention.